

Categorizing User Sessions at Pinterest

Dorna Bandari
Pinterest
dorna@pinterest.com

Shuo Xiang
Pinterest
sxiang@pinterest.com

Jure Leskovec
Pinterest
jure@pinterest.com

ABSTRACT

Different users can use a given Internet application in many different ways. The ability to record detailed event logs of user in-application activity allows us to discover ways in which the application is being used. This enables personalization and also leads to important insights with actionable business and product outcomes.

Here we study the problem of user session categorization, where the goal is to automatically discover categories/classes of user in-session behavior using event logs, and then consistently categorize each user session into the discovered classes. We develop a three stage approach which uses clustering to discover categories of sessions, then builds classifiers to classify new sessions into the discovered categories, and finally performs daily classification in a distributed pipeline. An important innovation of our approach is selecting a set of events as long-tail features, and replacing them with a new feature that is less sensitive to product experimentation and logging changes. This allows for robust and stable identification of session types even though the underlying application is constantly changing. We deploy the approach to Pinterest and demonstrate its effectiveness. We discover insights that have consequences for product monetization, growth, and design. Our solution classifies millions of user sessions daily and leads to actionable insights.

KEYWORDS

Web usage mining, Session categorization, Clustering

ACM Reference format:

Dorna Bandari, Shuo Xiang, and Jure Leskovec. 2017. Categorizing User Sessions at Pinterest. In *Proceedings of , Halifax, Nova Scotia, August 2017 (KDD'17)*, 9 pages. DOI: 00000

1 INTRODUCTION

As users interact with Internet applications a constant stream of event logs gets recorded. The ability to collect detailed event logs of user behavior presents the opportunity to gain important insights into the usage patterns of the applications, and also enables product personalization and recommendations. Particularly significant is the problem of user session categorization, where the aim is to classify types of sessions based on usage patterns.

User session categorization is an important problem as it leads to critical and actionable insights about the application. For example, discovering that a user commonly uses the application for casual browsing versus searching enables creation of a personalized home page, with buttons or links that make navigation easier. Additionally, using the information about users' previous behavior, applications can personalize recommendations, search ranking, as

well as advertising. Furthermore, categorizing sessions has significant value for business insights and strategy. Discovering session categories of an application and their relative scale leads to insights, such as estimating the strategic and monetary value of different use cases of the application, tracking changes in user behavior over time, and analyzing effects of product experiments. Such insights can guide the company in setting the right business and product goals as well as deciding which metrics to track and optimize.

There exists two main challenges in categorizing user sessions. First is the large scale and high dimensionality of the underlying data. User sessions often contain thousands of actions in a single session, such as clicking on various links, scrolling, searching, etc. Many of these actions may be unique. Moreover, an application may have millions of unique new sessions every day [1]. This makes discovering and categorizing patterns in user behavior extremely difficult, especially in applications with diverse and complex use cases (e.g. social networks, content discovery platforms, messaging platforms, and lifestyle applications). The second challenge is that user logs tend to be a constantly changing and unreliable data source [2]. As new product experiments are rolled out and the user interface changes, logged events tend to change. Therefore it is challenging to build a model on this dataset that will remain stable over time through most experiments and logging changes.

The problem of analyzing user sessions using event logs and clickstream data in online computing applications has been studied in the field of *web usage mining* [3–7]. Popular solutions in web usage mining include association rule learning and sequential pattern mining. However, these approaches were proposed for finding patterns and rules, while we focus on broad categorization of user sessions. Additionally, the majority of the solutions are too computationally complex to be deployed to large-scale Internet applications. Moreover, these methods generally do not address the problem of stability in the face of logging changes and product experimentation.

Present work. Here we develop a three-step approach for classifying usage sessions based on user event logs. Our solution has been deployed at Pinterest, a content sharing platform with over 150 million monthly users [8], and classifies tens of millions of user sessions per day. Additionally, we have evaluated our methodology on over six months of data and our method achieves stability in the face of logging changes and product experimentation. The three steps to our solution are as follows:

- We cluster a sample of sessions and create a labeled dataset for this sample.
- We use this labeled data to build a predictive model that is designed to achieve stability over time.
- We deploy the model in a distributed computational pipeline that performs daily scoring of all Pinterest user sessions.

Our methodology offers the following benefits: Our clustering method decreases the complexity of the problem without loss of important information. Instead of considering the session as a sequence of events in time, we consider each user session as a document, and each user action as a word in the document. We then apply document clustering methods to user sessions, which results in stable clusters that reveal major per-session use cases of Pinterest. Our predictive model focuses on the most robust and fundamental user events on the application, and it is not affected by logging changes and product experimentation. For example, logging a temporary new event on the application will not change the results of our daily classification. This is done by dividing user events into two groups, *Scoring features* and *Long tail features*, where Scoring features include only fundamental actions on the application. We define a new feature called *Noise feature* that replaces the set of Long tail features, decreasing sensitivity of our model to changes in them. Our classifier achieves more than 85% prediction accuracy.

After deploying the models on Pinterest, we identified six major session categories, each corresponding to a distinct and interpretable use case. We analyzed a sample of user sessions in the month of July 2016 (8.2 million sessions), and present a selection of insights. Our analysis revealed important patterns in user behavior. We found that content consumed by users in different session categories varies significantly, with low-intent session categories involving content that are aspirational or merely entertaining, and high-intent session categories involving highly practical content. We found that newer cohorts on Pinterest differ from older cohorts in session categories they utilize. Additionally, advertisement revenue differs substantially in different session categories, identifying opportunities for increasing revenue. We also present the difference in length of sessions and transition probability of session categories.

The rest of the paper is organized as follows. In Section 2 we will list existing solutions for web usage mining. In Section 3 we will describe the specific problem we aim to solve, and the associated dataset. In Section 4 we will describe our proposed solution in detail, with Section 4.2 describing cluster discovery, Section 4.3 detailing the design of the prediction models, and Section 4.4 describing our system in production. Section 5 lists the experiments and analyses that informed our design choices. In Section 6 we list specific details of our final deployed models, and illustrate some of the key insights we found from this work.

2 RELATED WORK

Web usage mining or clickstream mining has been an active area of research [9, 10]. Data mining algorithms have been applied to the user sessions for a variety of purposes, such as personalization of algorithms and applications [4, 5], recommendation systems [6], as well as obtaining business intelligence and aiding strategy [11]. Most of the research lies in one of two broad areas: association rule learning, which generally ignores the ordering between pages or actions, and path mining methods (including sequential pattern mining and clustering), which generally takes the order into account.

Association rule learning involves algorithms that discover rules that define relationships between items in a set (e.g., web pages) [9, 12]. An example would be discovering that users who visit a web

page *A* and take a specific action on that page are most likely to also visit web page *B*. Commonly the order of pages or actions is ignored. This class of algorithms have a variety of practical use cases, such as market and risk management and web personalization [13, 14]. The drawback of association rule mining methods is that the rules will not result in broad general categorization of sessions, while our method is used to find major user behavior categories.

Sequential pattern mining methods are the class of algorithms that discover patterns in a sequence of items [15], which in the case of web mining could be clicks or pages that users navigate to. The patterns may be found based on frequency of occurrence or some other measure of importance of a sequence. These methods are mainly practical for simple applications or parts of an application, such as a purchase funnel in an e-commerce application where the possible paths or actions are limited. Our method is designed specifically for large scale applications with complex navigation paths and cycles.

Clustering methods involve creating a set of features from the path a user takes as they navigate the web site. Shahabi et al. use time spent on each page as the main feature for the clustering [16]. Fu et al. categorize pages on a web site using an attribute-oriented approach [17] in order to decrease the dimensionality prior to clustering the sessions using hierarchical clustering [18]. Heer et al. use a combination of features of the pages, such as TF-IDF [19] of the content, as well as the path in order to find session clusters [20]. These methods were proposed for use on web site pages and links, rather than minute user actions, therefore their computational complexity would make their use for clickstream data in the scale of modern applications prohibitively large. Our method on the other hand is built for applications with millions of daily user sessions.

Jin et al. [21] propose using Probabilistic Latent Semantic Analysis (PLSA) to analyze web usage, and Xu et al. use PLSA to group web pages [22]. These methods are closest to our work, in that they reduce the dimensionality of the logging data by creating a matrix of sessions by unique web pages (or click actions). In [21], the matrix consists of only binary values, while in [22], time spent per web page is used to create the data. Neither of these methods address the issues of instability of the results in face of product experimentation and logging changes, while our system involves practical solutions to address this common problem.

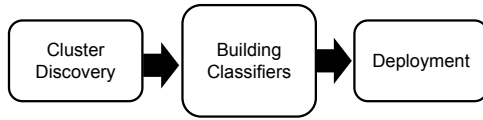
3 PROBLEM STATEMENT

When a user interacts with an Internet application, their actions are logged in the order they were performed. The aim then is (1) to discover types/classes of user behavior using event logs, and (2) to consistently categorize all usage sessions into the discovered classes. The focus of this paper is behavior in a single usage session so that categorization of all sessions can be performed on a daily basis.

There are several unique challenges and considerations when building a production system for session categorization. First is that modern Internet applications are complex and are available on multiple platforms. For example, Pinterest is available on six different platforms (iPhone, iPad, Android mobile, Android tablet,

Table 1: Example of four sessions and the list of events associated with each session

ID	User events in session
1	SEARCH, SEARCH, CLICKTHROUGH, CLICK, CLICK, SCROLL, SCROLL, SEARCH, VIEW, SEARCH
2	CLICK, PIN_VIEW, TAP, SEND_MESSAGE, READ_MESSAGE, SEND_MESSAGE
3	PIN_VIEW, REPIN, REPIN, SCROLL, TAP, REPIN, SCROLL, SCROLL
4	PROFILE_VIEW, PIN_VIEW, PROFILE_VIEW

**Figure 1: System overview**

web, and mobile Web) with each having distinct user interface elements and some distinct user actions.

Second, on Pinterest, like most complex Internet applications, there are several main activities a user can take part in. Users can view as well as save pieces of content which we call *pins*. They can click on a pin to visit the original web site that content links to, and they can search Pinterest for specific content. They can message each other, comment on pins, and read their notifications. In addition to these major activities, there are hundreds of other possible minor actions and user interface interactions a user can engage in. The challenge here is that there is no *a priori* clear way of knowing which of these actions are important to determine usage types and which ones can be discarded as noise.

4 METHODOLOGY

Our proposed solution consists of three stages (Figure 1). First, we discover session clusters using a sample of daily sessions. Next, we build a classifier that classifies each session into one of the clusters. Finally, we deploy the model in a distributed computational pipeline.

4.1 Data Preprocessing

First we process the raw event logs in order to assign each event to the unique user who performed the action, clean the data to remove spam and bot data, and filter out background event logs (i.e. events that are not triggered explicitly by user action, but by our application), and remove sessions that are too short to have had a purpose. A discussion of data preprocessing steps is beyond the scope of this paper.

We then sessionize the logs. There are a few different ways to define a user session [23]. In this work we take a time-lapse approach to defining user sessions, meaning that a user session is defined as a group of user actions that occur in an isolated period of time, with a pre-defined gap of inactivity before and after. This gap is found using the distribution of inter-action times on each device.

The cleaned, preprocessed data will have the form given in Table 1, with each row corresponding to a specific user session.

4.2 Cluster Discovery

In order to discover session-wise use cases we propose to cluster sessions based on user actions. We consider the list of user actions in a session as a document, with each user action as a word in the document. We use a sample of sessions in one day, and filter the actions to remove the ones that occur in fewer than 5% of sessions. We then find the TF-IDF of the actions [19]. In TF-IDF representation, the number of times an action occurs in a session is normalized by the inverse document frequency (IDF). IDF reduces the weight of more frequent actions in sessions. This means common actions such as *CLICK* or *SCROLL* in a mobile session would have lower weight in the clustering.

We normalize the vector of weights for each user session. The purpose of the normalization is to characterize each session by the actions that are dominant in that session, irrespective of session length.

We then use principal component analysis (PCA) to reduce the dimensions of the data, and cluster the projected results using K-Medoids [24] algorithm. Experiment results are listed in Section 5.1.

Summary of the cluster discovery method is as follows:

- (1) Sample user sessions in one day.
- (2) Filter out events that occur in fewer than 5% of sessions.
- (3) Find the TF-IDF weights for user events in every session.
- (4) Normalize the vector in each session.
- (5) Reduce the dimensions using PCA.
- (6) Cluster the projected results using K-Medoids.

4.3 Building Classifiers

The next step in our method is to build a multiclass classifier that will be deployed in production in order to assign session categories to daily user sessions. Note that session categories can be defined such that they map exactly to each session cluster found in Section 4.2, or alternatively, some clusters can be combined together to define a major session category. In our deployed system, after qualitative assessment of session clusters, we combined the clusters into 7 major session categories that correspond to major use cases of Pinterest. This simplification also made the categories consistent across different Pinterest applications.

One of the main reasons for building classifiers is achieving stability over time in spite of product experimentation and changes in user interface. Our classifier overcomes this issue by using a smaller subset of events that tend to be most stable in time.

Scoring features versus long tail features. User events are not equally robust over time. For example, events associated with minute user interactions with user interface elements may vary with minor design changes. E.g. changing the size of some elements on the page may change the frequency of triggering of another event. These changes may not be the result of fundamental changes in user behavior. The stability issue is caused by the following facts:

- Minor user interface changes can change the proportion of logged events, possibly leading to new erroneous clusters.

- When a new feature is being experimented on, it affects a small subset of users. This leads to artificially high IDF for user events associated with the new feature, as they seem rare. This could lead to new false clusters.

We aim to retrain the model once every three months. Consequently, we must choose a subset of events that would remain relatively steady in a three month period. However, note that we cannot assume stability of events that were not involved in past experiments and logging changes, since they may be affected by future ones. For this reason, it is best to only select fundamental actions on the application using prior knowledge, and only select additional events if prediction accuracy is not satisfactory. In other words, we aim to choose as few events as possible, while having acceptable prediction accuracy for every session category.

Another method for selecting the scoring features is by analyzing the historical daily counts of events. In this case, we would detrend and deseasonalize the daily counts of each event over a three month period, and find the variance of errors. Then, we would select events in the decreasing order of variance, stopping when an acceptable prediction accuracy occurs.

We name the chosen, stable events the *Scoring features*, and the rest the *Long tail features*.

Noise feature. Since we remove the long tail events from the prediction, we lose information in long tail user behavior, which we have labeled the *Noise* category. Therefore we have very poor accuracy in the *Noise* category, as demonstrated in Table 4. In order to improve the accuracy in this class, we will create a new feature called *Noise feature*. It is defined as

$$\eta_i = \frac{\sum_{f \in F_s} w_i(f)}{\sum_{f \in \{F_s \cup F_{lt}\}} w_i(f)}. \quad (1)$$

F_s is the set of scoring features, F_{lt} is the set of long tail features, and $w_i(f)$ is the TF-IDF weight of feature f in session i . Since long tail events are possibly changing drastically over time, we normalize the η_i feature by the mean of this feature over all sessions. This step is important in ensuring that a sudden change in weight of some of the long tail features will not create large shifts in size of session categories.

Finally, using the Scoring features along with the Noise feature, we build a random forest model to predict the session categories. Prediction accuracy of the models trained with and without the new Noise feature is given in Table 4 in Section 5.2.

Summary of the classification step is given below.

- (1) Start with the labeled dataset from clustering method, a sample of user sessions with session category labels.
- (2) Find TF-IDF of all events.
- (3) Create the Noise feature using Equation 1, and normalize it across all sessions.
- (4) Normalize the vector of scoring features in each session.
- (5) Train a random forest model using data from step 2 and 3.

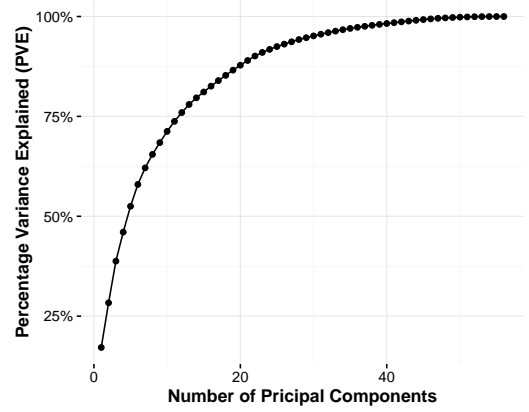


Figure 2: On iPhone devices we selected 14 principal components, which explains 80% of variance in the data.

4.4 Deployment

Having considered the downstream workflows as well as Pinterest’s own computation infrastructure, we decide to use two separate pipelines to accommodate computational tasks that are of various deployment frequencies. Specifically, we have a daily pipeline that performs the aforementioned data preprocessing and scoring, since the results of both tasks will be consumed daily by downstream jobs. On the other hand, cluster discovery and model update are currently carried out on demand. Though it is possible to combine these tasks into a single workflow, we believe the current setup provides better balance between development velocity and scalability.

Daily scoring. Once a model is obtained, it is deployed online to compute scores for each session in our entire context session logs. Due to the large amount of daily session data, a distributed computational pipeline is necessary and we implement an end-to-end scoring pipeline in Apache Spark. We use Pinball¹ to compose a workflow that will be scheduled daily.

IDF. Rarity of events may change as product experiments are rolled out to different proportion of user base. This means that even user sessions that are not part of the experiment would be affected by the experiment, since the IDF is calculated over the entire data set. So while in this case we cannot do much about the user sessions in the experiment, we should ensure that session not in the experiment will not be affected. Therefore, given that the prediction model will take TF-IDF weights of the subset of events, the IDF should not be part of the computation pipeline, and it should only be updated every time the model is re-trained.

Updating the model. Model update usually comes with recomputing or back-filling existing data. To better support retrospective experimentation, we have stored versioned data on a persistent storage system and by default show the latest result.

Table 2: Clustering trials on the iPhone Pinterest application. We chose 13 clusters, since clusters are found to be stable, and Noise cluster is smaller than the case with 8 clusters.

Number of clusters	18	13	8
% in Noise cluster	9%	15%	21%
Avg. Stability, Jaccard distance	0.91	0.89	0.96
Avg. Silhouette	0.3	0.33	0.39
% Clustered with stability of +0.7	96%	100%	100%

5 EXPERIMENTS

In this section we present the results of some of the experiments that lead to some of our important design choices. Note that some of the choices were made through qualitative assessment of the results by the authors and our internal partners. This is an important step in implementing such a system for practical applications.

5.1 Evaluating Clustering

We select the number of principal components such that more than 80% of variance is explained by the projected features. Figure 2 illustrates the Percentage Variance Explained (PVE) plot for the iPhone Pinterest application. The elbow in this plot occurs at around 80% value, therefore we choose 14 principal components in this case.

In order to validate the clustering method and select the right number of clusters, we considered a few different factors. One is qualitative assessment and usability of clusters by our internal partners. Our internal conversations lead us to choose broad types of clusters as opposed to smaller clusters. Additionally, we considered the percentage of sessions that were clustered in the *Noise cluster*, with the aim of decreasing the size of this cluster. Other metrics we assessed are Silhouette score [25], which measures the gap between clusters, and cluster stability [26, 27], which measures reproducibility of clustering results. For cluster stability, we created 50 randomly selected subsets of the data set, each having half of the original data, as suggested in [27]. We found the average Jaccard similarity of each original cluster membership to the one found in each subset. We assumed a cluster is stable if more than 70% of the points were assigned to the original cluster.

Table 2 summarizes three of the clustering options we considered, along with the metrics noted above. We chose 13 clusters in this case, since all data are in stable clusters, and size of the Noise cluster is not very large.

5.2 Evaluating Classifiers

The classifiers assign session categories to each user session. The clusters from previous step are combined into 7 major session categories. This was done for simplification by qualitative assessment of the clusters, as well as to unify the session categories across different Pinterest applications.

We built each model on 500K sessions on a single day, with 1/3 of data assigned as the test set. Table 3 shows the confusion matrix for the final classifier for the iPhone application.

¹ <https://github.com/pinterest/pinball>

Table 4 lists the misclassification error of the multiclass random forest classifier in iPhone application for three cases: with all user events, with only Scoring features, and with Scoring features plus the Noise feature (Eq. 1). Note that the Noise cluster has a much higher error rate without the new feature. Overall, total classification error in test data was found to be 9.7% in this device.

In every device we built a classifier with less than 15% total classification error. The class with the largest error in every case was the Noise cluster, which is expected, given the features we have removed were the long tail events that were contributing to sessions in this cluster. Given that by definition this class is not very important in our understanding of sessions, the added error in this class is a good trade-off for the added stability.

Figure 3 illustrates the daily proportion of each session category in a three month period in 2016. Most session category names are anonymized for confidentiality reasons, with the exception of Noise and Clickthrough categories. This plot demonstrates stability of the classification results in face of numerous product experiments and user interface changes that are regularly conducted. Additionally, we have the entire data available for over 6 months of activity on Pinterest and have confirmed that the results are robust in time.

6 APPLICATION TO PINTEREST

In this section we present an application of our methodology to Pinterest data [28, 29]. We created a different set of models for each Pinterest application (iPhone, iPad, Android mobile, Android tablet, web, and mobile Web). In each case, we built models using a sampled set of 500K user sessions on a single day.

For cluster discovery, we used 55 user actions. We found a different number of clusters in each application, e.g. 13 clusters on the iPhone application. After qualitative assessment, we combined the clusters on all applications into 7 major session categories. These session categories are consistent across all devices, and correspond to major session-wise Pinterest use cases. They are named using qualitative assessment of weight of events in each session cluster. The list is as follows: Browse (i.e. mainly viewing content), Clickthrough (i.e. clicking to the web site that the pin links to), Notification (i.e. interacting with the in-application notifications), Retrieval (i.e. viewing content previously saved), Repin (i.e. saving content), Search (i.e. searching pins), and Noise (i.e. long tail use cases).

We chose a subset of 12 events as scoring features in the classifiers, as described in Section 4.3. The random forest models were built using the randomForest package in R [30], with 250 trees and minimum leaf size of 1000 rows. The models were converted to Predictive Model Markup Language (PMML) [31] to be deployed in our distributed pipeline for daily scoring.

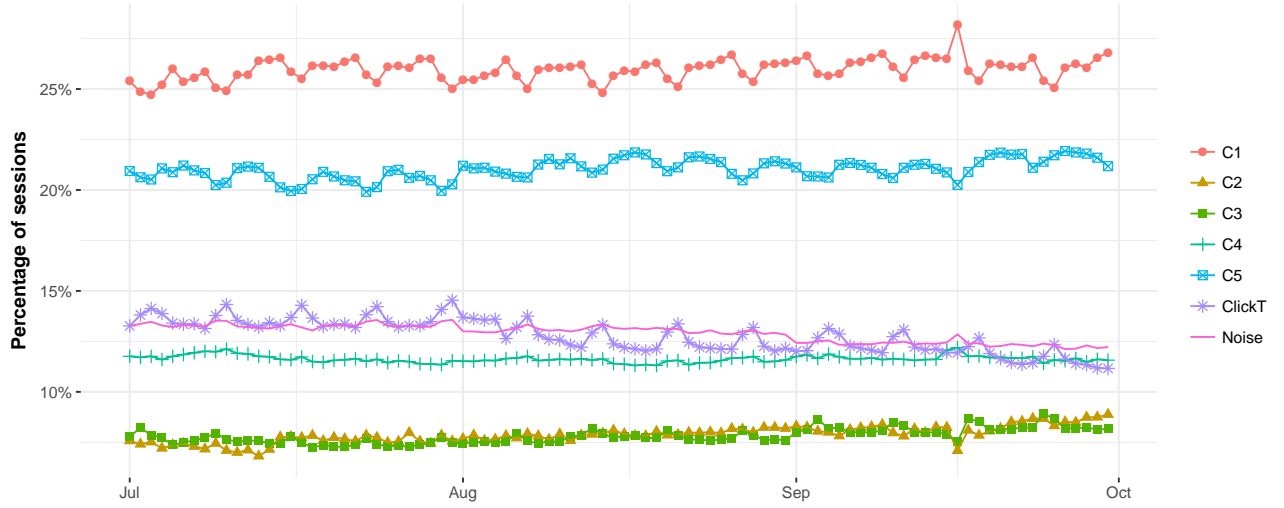
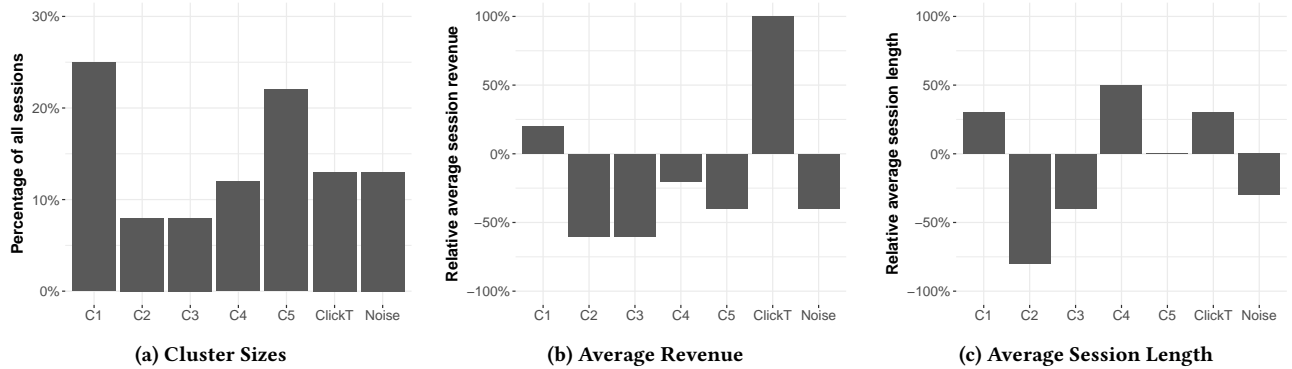
In the following section we present insights found using this work. Due to confidentiality considerations we anonymized session category names in most of the plots, and refer to them by ID instead. The only session categories we will have in every plot are Noise and Clickthrough sessions.

6.1 Insights

The diversity and the number of major use cases of Pinterest was one of the major insights from this work. Here we present some

Table 3: Confusion matrix for final classifier for Pinterest iPhone application, on the test dataset. The session categories are named using qualitative assessment of the data.

Category	Browse	Clickthrough	Notification	Noise	Retrieval	Repin	Search
Browse	93.9%	0.9%	0.2%	2.5%	0.7%	1.0%	0.7%
Clickthrough	0.8%	94.2%	0.1%	2.1%	1.2%	0.6%	1.1%
Notification	0.4%	0.1%	97.2%	1.5%	0.5%	0.2%	0.0%
Noise	8.1%	2.2%	1.1%	73.2%	9.5%	3.9%	2.0%
Retrieval	2.6%	2.1%	0.3%	9.5%	82.0%	1.8%	1.8%
Repin	2.1%	0.4%	0.2%	3.4%	0.9%	92.0%	1.1%
Search	0.6%	0.3%	0.1%	0.8%	0.3%	0.8%	97.2%

**Figure 3: Daily patterns of session categories in a three month period in 2016. Each session category follows a regular weekly seasonal pattern in this period, in spite of hundreds of product experiments on the application. Long term trends are mainly due to annual seasonality. Note that session category names are anonymized for confidentiality reasons, with the exception of Noise and Clickthrough categories.****Figure 4: (a) Cluster sizes vary from 8% to 25%. 13% of sessions are not classified, i.e. are in Noise category. (b) Revenue varies on different session categories. (c) Average session length in each session category, compared to overall average session length.**

of these insights that were enabled by our session categorization methodology.

First, Figure 4a illustrates the size of each cluster across all devices in one month of data. The smallest clusters are 8% of sessions,

Table 4: Misclassification error rate in each class, when classifier uses only the subset of robust events, the subset of robust events with the Noise feature, and all events. The middle column has much lower error in the Noise category, while achieving robustness over time by eliminate the more volatile features.

Cluster	Scoring features only	Scoring features + Noise feature	Scoring + Long-tail features
Browse	6%	6%	%3
Clickthrough	6%	6%	%2
Notification	3%	3%	%1
Noise	80%	27%	%7
Retrieval	20%	18%	%5
Repin	7%	8%	%5
Search	2%	3%	%2

Table 5: Highest average feature weights in Search and Clickthrough session categories

Search Category	Clickthrough Category
SEARCH_VIEW, 7.6	LOAD_URL, 8.5
SEARCH_PINS, 3.0	BROWSER_VIEW, 2.8
VIEW_END, 1.6	PIN_CLICKTHROUGH, 2.4
DISCOVER_VIEW, 1.4	VIEW_BEGIN, 1.5
VIEW_BEGIN, 1.2	VIEW_END, 1.5
LOAD_URL, 1.1	PIN_VIEW, 0.98

and the biggest ones are more than 20% of sessions. Note that 13% of all sessions are long tail forms of engagement, which we labeled Noise session category.

Second, we make an observation in Figure 5a, which demonstrates the daily proportion of session categories in the month of July 2016. The proportion of some session categories exhibit weekly seasonality, with Clickthrough and C4 session categories having the opposite weekly pattern to C1 and C5.

And last, Table 5 lists the highest weighted features in Clickthrough and Search categories. In each case, we list the highest weighted features (i.e. user events) that are associated with each form of engagement. We observe that clusters vary a lot in terms of events that are important in each. For example, Search category has a high average weight for SEARCH_VIEW event while the Clickthrough category has a high weight for LOAD_URL event. This demonstrates how the category names were selected.

Next we shall discuss further insights obtained by our clustering methodology in more detail.

Cohorts. First we examine how Pinterest usage varies among different user cohorts. Figure 5b displays the proportion of each session category by user’s cohort, i.e. the year they signed up for Pinterest. We observe that older cohorts and newer cohorts are similar in the proportions of session categories C2 through C4, as well as Clickthrough sessions. However, interestingly session category C1 is utilized much less by older cohorts, and session category C5

is used much more by them. This shows how user behavior has changed on Pinterest as a response to changes in the application as well as cohorts. This finding enables the company to plan for more forward looking product features and functionality.

Revenue varies by session category. Next we also examine the amount of revenue generated by sessions of different categories. Figure 4b illustrates the normalized value of revenue per session, by session category. The values are normalized by average per session revenue. We find significant variation of revenue between different session categories. In particular, Clickthrough sessions and C1 have significantly more revenue per session than others. This finding has important implications in setting company strategy, e.g. in deciding which session-wise use case requires more investment in product development.

Content. Another interesting aspect we examine is how the content varies across different session categories. For a specific category of content, namely Food category, we compared pins that were viewed or interacted with on different session categories. We found that pins that were more likely to have been interacted with in Clickthrough sessions are practical recipes and specific ingredients (+10% more likely, with $p < 0.05$). On the other hand, pins with beautiful pictures, desserts, and aspirational content are more likely to have been engaged with in Repin session category (difference is +10%, with $p < 0.05$). We observed similar patterns in other content categories as well, where pins with practical content are more likely to be viewed in Clickthrough sessions, and aspirational content in Repin sessions.

Session length. We also observe that session categories differ significantly by length, which is a metric commonly used to understand depth of engagement. This enables product development to focus on increasing depth of engagement for specific session-wise use cases of the application. This is illustrated in Figure 4c.

Thanksgiving daily patterns. Our clustering methodology also enables us to “zoom-in” on a particular timeframe to better understand how holidays shape the usage of Pinterest. In Figure 6 we compare the daily counts of the Clickthrough and Search session categories in United States in November 2016, normalized by average daily sessions of each category. The daily patterns around Thanksgiving holiday shows that Clickthrough sessions have a sharp increase on this holiday, which traditionally involves cooking a meal with the extended family. On the other hand, the number of Search sessions increases a few days prior to the holiday, as users start planning for the holiday.

Transition probabilities. Last we examine how users transition between sessions of different categories. The question here is whether users tend to use Pinterest in only a single way or whether there are many different session categories a single user engages in.

Transition probabilities between session categories are illustrated in Figure 7. Edges with transition probability less than 0.15 were removed for simplification. Observing the probability of getting the same session category in consecutive sessions, we can see that some session categories are much more likely to be repeated consecutively than others. Additionally, session category C1 is the most likely session to follow all other session categories.

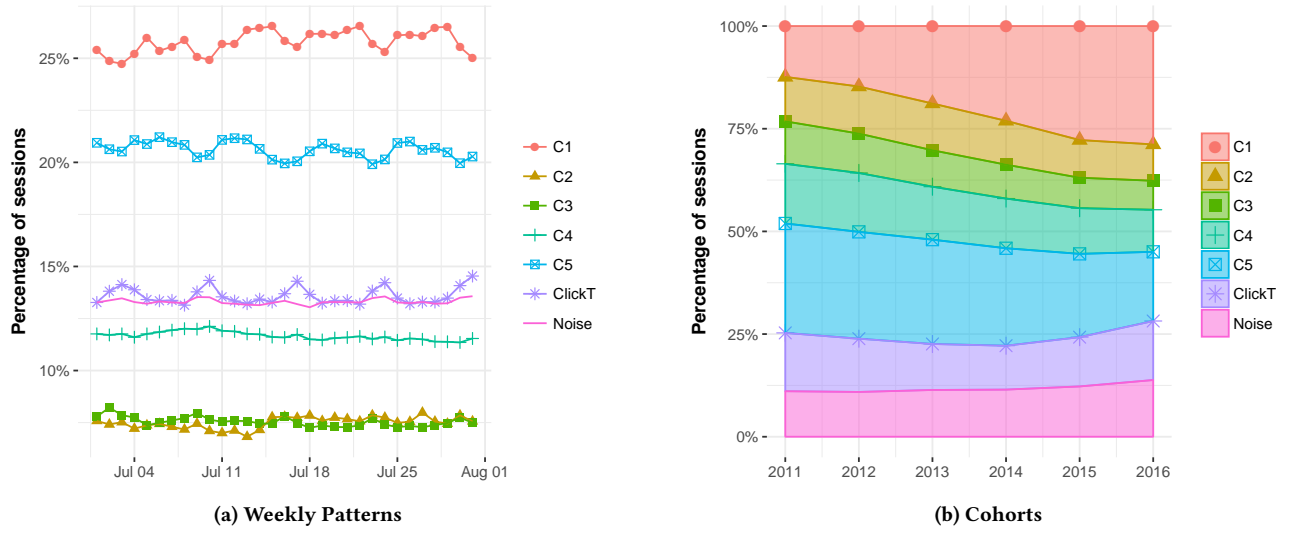


Figure 5: (a) Proportion of sessions in month of July. C3 and Clickthrough session categories have opposite weekly patterns to C1 and C5, with C3 and Clickthrough peaking on weekends, and C1 and C5 peaking mid-week. (b) Different cohorts have different proportion of session categories, with C5 being used less by newer users and C1 being used a lot more.

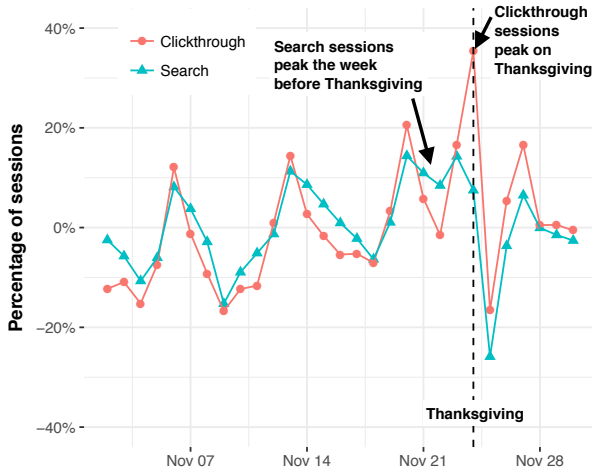


Figure 6: Comparing indexed counts of Search and Clickthrough session categories on and around Thanksgiving holiday in U.S. Count of Search sessions increase a few days prior to the holiday, whereas Clickthrough sessions have a sharp pick on the day.

7 CONCLUSION

In this paper we studied the problem of user session categorization, where the goal was to automatically discover categories/classes of user in-session behavior using event logs, and then consistently categorize each user session into the discovered classes.

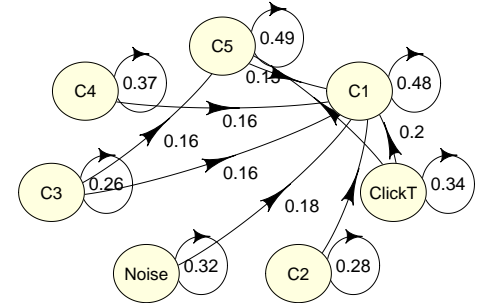


Figure 7: Transition probability of session categories. For simplification the edges with less than 0.15 probability have been removed.

We developed a three stage approach which uses clustering to discover categories of sessions, then builds classifiers to classify new sessions into the discovered categories, and finally performs daily classification in a distributed pipeline. An important innovation of our approach was defining a Noise feature that allowed for robust and stable identification of session categories, even though the underlying application is constantly changing. We deployed the solution at Pinterest where it classifies millions of user sessions daily, and provides actionable insights.

Future work could investigate predicting session categories using only the first few actions which would enable personalized experiences within a single visit. Furthermore, it would be interesting to monitor which user demographic subgroups use the product

in a given way, and subsequently connect changes in usage patterns to releases of new product features.

8 ACKNOWLEDGMENTS

We thank Dan Lurie, Chunyan Wang, and Austin Chang for valuable insights and support throughout the project, Tien Nguyen for help with deployment of the system, and Grace Huang, Dan Frankowski, Brian Karfunkel, Roja Bandari, and Minli Zang for their valuable discussions and insights.

REFERENCES

- [1] Gang Kou and Chunwei Lou. Multiple factor hierarchical clustering algorithm for large scale web page and search engine clickstream data. *Annals of Operations Research*, 197(1):123–134, 2012.
- [2] A. Katal, M. Wazid, and R. H. Goudar. Big data: Issues, challenges, tools and good practices. In *2013 Sixth International Conference on Contemporary Computing (IC3)*, pages 404–409, Aug 2013.
- [3] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43(8):142–151, August 2000.
- [4] Magdalini Eirinaki and Michalis Vazirgiannis. Web mining for web personalization. *ACM Trans. Internet Technol.*, 3(1):1–27, February 2003.
- [5] Tak Woon Yan, Matthew Jacobsen, Hector Garcia-Molina, and Umeshwar Dayal. From user access patterns to dynamic hypertext linking. In *Proceedings of the Fifth International World Wide Web Conference on Computer Networks and ISDN Systems*, pages 1007–1014, Amsterdam, The Netherlands, The Netherlands, 1996. Elsevier Science Publishers B. V.
- [6] Yoon Ho Cho, Jae Kyeong Kim, and Soung Hie Kim. A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications*, 23(3):329–342, 2002.
- [7] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: Discovery and applications of usage patterns from web data. *Acm Sigkdd Explorations Newsletter*, 1(2):12–23, 2000.
- [8] Ben Silbermann. 150 million people finding ideas on pinterest. *Pinterest Blog*, 2016.
- [9] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.
- [10] R. Cooley, B. Mobasher, and J. Srivastava. Web mining: information and pattern discovery on the world wide web. In *Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence*, pages 558–567, Nov 1997.
- [11] S. Tiwari, D. Razdan, P. Richariya, and S. Tomar. A web usage mining framework for business intelligence. In *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 731–734, May 2011.
- [12] Jinze Liu, S. Paulsen, Wei Wang, A. Nobel, and J. Prins. Mining approximate frequent itemsets from noisy data. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4 pp.–, Nov 2005.
- [13] Michael J. Shaw, Chandrasekar Subramaniam, Gek Woo Tan, and Michael E. Welge. Knowledge management and data mining for marketing. *Decis. Support Syst.*, 31(1):127–137, May 2001.
- [14] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd International Workshop on Web Information and Data Management, WIDM '01*, pages 9–15, New York, NY, USA, 2001. ACM.
- [15] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering, ICDE '95*, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society.
- [16] C. Shahabi, A. M. Zarkesh, J. Adibi, and V. Shah. Knowledge discovery from users web-page navigation. In *Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE '97) High Performance Database Management for Large-Scale Applications, RIDE '97*, pages 20–, Washington, DC, USA, 1997. IEEE Computer Society.
- [17] Jiawei Han, Yandong Cai, and Nick Cercone. Knowledge discovery in databases: An attribute-oriented approach. In *Proceedings of the 18th International Conference on Very Large Data Bases, VLDB '92*, pages 547–559, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.
- [18] Yongjian Fu, Kanwalpreet Sandhu, and Ming-Yi Shih. A generalization-based approach to clustering of web usage sessions. In *Revised Papers from the International Workshop on Web Usage Analysis and User Profiling, WEBKDD '99*, pages 21–38, London, UK, UK, 2000. Springer-Verlag.
- [19] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.
- [20] Jeffrey Heer, Ed H. Chi, and H. Chi. Mining the structure of user activity using cluster stability. In *Proceedings of the Workshop on Web Analytics, SIAM Conference on Data Mining (Arlington VA. ACM Press, 2002.*
- [21] Xin Jin, Yanzan Zhou, and Bamshad Mobasher. Web usage mining based on probabilistic latent semantic analysis. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04*, pages 197–205, New York, NY, USA, 2004. ACM.
- [22] Guandong Xu, Yanchun Zhang, and Xiaofang Zhou. Using probabilistic latent semantic analysis for web page grouping. In *15th International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications (RIDE-SDMA'05)*, pages 29–36, April 2005.
- [23] Myra Spiliopoulou, Bamshad Mobasher, Bettina Berendt, and Miki Nakagawa. A framework for the evaluation of session reconstruction heuristics in web-usage analysis. *INFORMS J. on Computing*, 15(2):171–190, April 2003.
- [24] Leonard Kaufman and Peter Rousseeuw. *Clustering by means of medoids*. North-Holland, 1987.
- [25] Peter Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, November 1987.
- [26] Tilman Lange, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16(6):1299–1323, 2017/01/29 2004.
- [27] Christian Hennig. Cluster-wise assessment of cluster stability. *Computational Statistics and Data Analysis*, pages 258–271, 2007.
- [28] Caroline Lo, Dan Frankowski, and Jure Leskovec. Understanding behaviors that lead to purchasing: A case study of pinterest. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pages 531–540, New York, NY, USA, 2016. ACM.
- [29] Justin Cheng, Caroline Lo, and Jure Leskovec. Understanding behaviors that lead to purchasing: A case study of pinterest. In *Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee*, 2017.
- [30] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.
- [31] Alex Guazzelli, Michael Zeller, Wen-Ching Lin, Graham Williams, et al. Pmml: An open standard for sharing models. *The R Journal*, 2009.